

POCKET COMPUTER

NEWSLETTER



© Copyright 1982 — Issue 11

January

SHARP PREPARING NEW POCKET COMPUTER

Reliable sources report that Sharp Electronics, Inc., producer of the popular PC-1211 and Radio Shack TRS-80 versions of pocket computers, is just about ready to release an exciting, dramatically enhanced, new pocket computer. The upgraded unit will be designated the Sharp PC-1500.

Preliminary reports indicate that the new device will rival the capabilities of the recently announced RL-H1000 Hand Held Computer marketed by Panasonic.

For starters, the PC-1500 will have increased memory capacity. It is reported to contain 16 kilobytes of ROM memory containing an improved BASIC language and operating system plus approximately 3.5 kilobytes of user-programmable memory. Additionally, expansion units will allow more memory to be coupled to the system.

A specially-designed 8-bit CPU using CMOS technology is at the heart of the new unit.

The machine will be housed in a "pocketable" case that is slightly larger than the current PC-1211. Measurement figures of approximately 7 inches by 3.5 inches by 1 inch have been reported. A QWERTY keyboard arrangement that is somewhat more comfortable to use than the present PC-1211 version is also claimed, along with six special "programmable" keys.

BASIC language enhancements are said to include the ability to manipulate string variables and a built-in real-time clock.

The really big news, however, is that the unit has been specifically designed to interface with a variety of peripheral devices that Sharp will be introducing during 1982.

The first such add-on to be made available is reported to be a small printer that will be capable of producing print-outs in four different colors! This interface will also enable data to be stored and recovered using an audio tape recorder. Reports indicate that this interface will be referred to by the model designation CE-150.

Another peripheral will enable the PC-1500 to display information on a television set or video display.

Still other add-ons will provide additional memory and the ability to communicate with other devices.

For instance, it has been reported that it will be possible to simultaneously control two different tape units so that data can be read from one unit while it is stored to a second. This will permit relatively powerful data-handling procedures to be implemented. It has also been ascertained that the tape units might eventually be replaced by floppy disks or similar fast-access mass storage devices.

The PC-1500 will reportedly be available in the United States in late February or early March, 1982. A retail price of approximately \$300 has been indicated. As confirmation of this information, it has been reliably reported that the unit will be available in Japan at the start of 1982 at an equivalent U.S. price of \$277.00.

The CE-150 combination 4-color printer and tape cassette interface will probably have a U.S. price of approximately \$250.00.

No word has yet been obtained by PCN as to whether or not Radio Shack will market a version of the PC-1500. However, it is reported that Sharp Electronics, Inc., is forming a new U.S. marketing division to boost the sales of the PC-1500 and similar products.

ROUTINE PERMITS INDIRECT SPECIFICATION OF FILE NAMES

The accompanying routine enables an unskilled user to specify a file name when saving data. Subsequently, the user can call for that same data file to be recovered. This type of *indirect* file name specification eliminates the need for a user to be familiar with the actual I/O commands used by the Radio Shack TRS-80 and Sharp PC-1211 pocket computers.

The routine uses a technique submitted by *Kendal B. Stonebrook*, 811 Princeton Avenue, Lansing, MI 48915. The file name specified by the user is first stored in the highest unused array element available in memory. This element is then recorded as a separate data file having the name INDEX. Immediately following this file, another data file is recorded using the name FILE. This is the file that contains the user's actual data.

When a user-named file is desired, another routine reads each file named INDEX and examines the stored array element for the name specified by the user. When a match is found, the following file is loaded to retrieve the desired data!

Notes

Line 930 may be used to obtain the highest array element available in the PC as this will vary depending on the size of the installed program. (You can convert this line from the general case once the size of a program is determined.) Line 935 then places the user-defined file name in the array element A\$(C) and activates the tape unit to record this element in a data file named INDEX. Line 940 proceeds to record another file named FILE that contains the actual data elements that are being stored. A GOTO or RETURN statement may be appended to this line as appropriate for a specific application.

Recovery of a user-named file begins at line 900.

The routines illustrated use variables A, B, C, A\$(27) and A\$(NNN) where NNN is the highest array element available.

Program Indirect File Names

```
900: "F"INPUT "FI          +1: IF B>12
    LE NAME? ";A          PRINT "N/F":
    $:A$(27)=""           END
    B=0                   925:GOTO 905
905: INPUT # "INDE        930: "D"INPUT "#
    X";A$(27)             MEMORIES LEF
910: IF A$(27)=A$         T? ";C:C=C+2
    BEEP 4:PAUSE          6
    "LOADING "A$          935: INPUT "FILE
    $(27);" FILE          NAME? ";A$(C
    ":INPUT # "FI        ):PRINT # "IN
    LE":PRINT "F          DEX";A$(C):
    ILE LOADED":         BEEP 4
    GOTO "A"             940:PRINT # "FILE
920: PAUSE "SKIPP         ":PRINT A$(C
    ING ";A$(27)         "; "FILE OUT
    "; "FILE":B=B        "
```

PRODUCT REVIEW AVIATION PROGRAM

Produced by: *Radio Shack for the TRS-80 Pocket Computer.*

List price: \$24.95

Availability: *Radio Shack Computer Centers and selected Radio Shack stores.*

Reviewer: *Nat Wadsworth, Editor.*

Wow! I'll say it right off the top: Radio Shack has improved their PC software since the last time I took a good look.

This is their package of programs for aviators. Since nobody flying a private plane would want to try and load a program from cassette while attempting to maintain a vector under IFR (instrument) conditions, this package is designed so that all the routines reside in memory at one time. They did it with zero memories to spare!

The package is also designed to operate with or without the printer. You can thus work up a proposed flight using the printer and take the hardcopy as your initial flight plan. Once in the left seat you can work with just the PC fastened to your knee-clipboard. (A couple of pieces of Velcro will do a nice job of holding the PC to the board.)

There is a lot in this package. First, you have the standard conversions that a pilot frequently requires. These include Centigrade to Fahrenheit temperature and vice versa, statute miles/nautical miles/kilometers, U.S. gallons to Imperial gallons and liters, etc.

Then you have all the typical calculating functions that a pilot is likely to need both pre-flight and in-flight. These include: true altitude (using pressure altitude, temperature and a reference altitude), true airspeed (using pressure altitude, temperature and indicated airspeed), Mach number (calculated from true airspeed and temperature) or true airspeed (as a function of Mach number and temperature), off-course correction and drift angles, true heading and ground speed with winds and enroute winds. Note that there are Mach calculations for you Lear jockies!

Everything operates with the PC in the DEF mode. And, wonders of wonders, Radio Shack has even included a completely labeled keyboard template so that you don't have to try and remember which key gives you what function! Need to convert statute miles to nautical miles? Just press SHIFT/M just as it states on the template. Need a course correction? SHIFT/D says the template.

So, O.K., let's say we have been lost in the clouds for awhile and we discover we need to use SHIFT/D. The program pops up with DOC for "distance off course." Next it wants to know DF (distance flown) and DTF (distance to fly). You just pull those figures off the current leg of your flight chart and give them to the computer. In about three seconds the PC responds with an ATI (angle to intercept) figure. You bank her, watch the heading indicator come around and then level your wings on your new intercept heading. If you want, the PC will give you an ATP (angle to parallel your original course) and the drift angle (DA). The three seconds or less needed to perform the calculations isn't bad. It is less than I feared would be the case when I first saw the package.

Want to determine a true course heading on a new leg? Punch in SHIFT/H. The computer will ask for TC (true course), TAS (true air speed), WD (wind direction) and WS (wind speed). In just a few moments the computer will respond with GS (ground speed) and the TH (true heading) to steer on the leg. Simple. What a way to fly.

Yep, they really used their heads on this package. They even had the good sense to assign a separate variable to each of the parameters one generally has to juggle and/or keep track of during a flight. What is more, they even provide a complete list of the variables and their parameter assignments and point out (in the manual) how you can access and utilize the assignments in your own calculations.

The manual itself is only 20 pages, but it is sufficient to give any pilot the information needed to use the package. Once you have been through the manual and tried the procedures, you should be pretty well set just using the keyboard template as a guide.

Nice job on this one, Radio Shack. I give it a 9 on a scale of 10. Now, if they would do the same kind of job in areas that have broader application for the general public, people might get the idea that PCs can be really useful!

INSTRUCTION EXECUTION TIMES FOR THE RADIO SHACK TRS-80 POCKET COMPUTER

Are you trying to minimize program execution time? Speed can be significantly affected by the proper choice of variables, operations, and method of transfer. This is especially true when a portion of a program needs to be repeated a large number of times.

Norlin Rober has compiled the timing information presented here. Remember, the actual measurements shown may vary slightly on individual PCs. However, the relationships and relative speeds of the various types of directives will remain essentially the same for your unit.

Watch Those Program Transfers!

Perhaps the greatest time-saving technique is to shorten the search time when using GOTO and GOSUB statements. A transfer to a *line number* is relatively fast when the location sought is near the GOTO or GOSUB statement, *regardless of direction!* A transfer to a *label* is fastest when the label is located near the beginning of program memory.

Searches for labels or line numbers can be quite slow if the search must take place over a large number of program bytes. In some cases a single transfer can take as long as 2.7 seconds. If such a transfer is to be performed a large number of times (as, say, part of a loop), some relocation of program lines can significantly reduce execution time.

The exponentiation operation is also relatively slow, taking up to one second. Using AA in place of A^2, AAA in place of A^3, etc., does more than simply speed up operation. It also permits the use of negative values of A and often increases accuracy!

A Note about Timing Measurements

Execution times were obtained using the following method: The routine —

```
1 FOR X=0 TO 199
2 NEXT X
3 BEEP 1
```

takes 36.1 seconds to execute (on the test TRS-80 PC). The BEEP is used to assist in obtaining the actual timing. If line 2 is altered to:

```
2 W=A:W=A:W=A:W=A:W=A:W=A:NEXT X
```

execution takes 95.5 seconds. That is 59.4 additional seconds for the 1000 iterations of W=A. Thus, for purposes of the study, W=A takes 0.0594 seconds per execution.

Storage and Recall

Storage in variables A to V takes .017 seconds longer than storage in W to Z. However, recall of A to V takes .016 seconds less than recall of W to Z! The measurement for W=A was .059, for W=W it was .075, A=A took .076 and A=W was .092 seconds. Storage (into W) of a constant (entered by a program statement) takes from .055 to .150 seconds, depending on the number of bytes making up the constant. W= π takes .055 seconds.

If the number being stored is 0, then an additional .003 seconds is required.

Variables specified by subscript require an additional .030 seconds if the subscript has one digit. It takes .038 seconds if the subscript has two digits and .046 for a three-digit subscript.

Addition and Subtraction

Execution time depends largely on the number of shifts required to align the decimal points and on whether the first or second number has the larger exponent.

W=A+B or W=A-B takes from .089 to .300 seconds if the exponent of B is the larger, and from .089 to .367 seconds if the exponent of A is the larger. An additional .001 to .011 seconds is used if shifting is needed to normalize the calculated result.

A negative result requires an additional .002 seconds. A zero result takes an extra .003 seconds.

If variables having longer store or recall times are used, the execution times are modified accordingly.

Multiplication

Execution time depends primarily on the sum of the digits of the multiplier. W=AB takes from .091 to .153 seconds, with .005 seconds

additional if one of the factors is zero. $W=A*B$ takes .009 seconds longer than $W=AB$. If constants or variables having longer store or recall times are used, then execution times are modified accordingly.

Division

Execution time depends considerably on the sum of the digits in the twelve-digit internally calculated quotient. $W=A/B$ requires from .120 to .193 seconds. If constants or variables having longer store or recall times are used, then execution times are modified accordingly.

Logical Calculations

The logical calculations $A>B$, $A\geq B$, $A=B$ and $A<>B$ are evaluated by calculation of $A-B$. The time required is .002 to .003 seconds longer than the subtraction time.

The calculations $A<B$ and $A\leq B$ use the subtraction $B-A$. These take .003 to .004 seconds longer than the subtraction time.

Exponentiation

The calculation of A^B is performed using $10^{B*\text{LOG } A}$. Execution time is the combined time of the calculation of $\text{LOG } A$, multiplication by B and calculation of the required power of 10.

Thus, calculation time for $W=A^B$ is typically .094 seconds when $A=0$. It varies from .279 seconds to approximately 1.0 seconds when A is greater than zero.

Remarks

A REMark statement is bypassed in .012 to .146 seconds, depending on its length.

The GOTO Statement

The statement *GOTO line number* results in a search forward if the line referred to is greater than the current line number. The search is in the reverse direction if the line number is less than the current line number. When searching in the forward direction, execution of this GOTO directive takes .067 seconds plus .185 seconds per 100 bytes traveled. In the reverse direction the traversing time is .147 seconds per 100 bytes.

The statement *GOTO label* results in a search for the referenced line beginning at the start of program memory. This procedure takes .042 seconds plus .182 seconds for each 100 bytes searched. Additionally, each label preceding the one sought consumes .004 seconds of time.

The GOSUB Statement

The GOSUB directive requires .024 seconds longer than the corresponding GOTO statement. The extra time is required to store the return address and execute the RETURN statement. Note that the RETURN transfer does not have to search memory for an address.

The IF Statement

The statement IF A takes .051 seconds if A is positive and .064 to .200 seconds if A is not positive, depending on the length of the remainder of the line that contains the IF statement.

The statement IF (expression) takes .005 seconds less than execution of the statement $W=(\text{expression})$ if the expression is positive. Otherwise it takes .005 to .141 seconds longer.

Loops

The FOR statement using W , X , Y or Z (not requiring calculation to determine loop parameters) takes from .112 to .161 seconds, with .040 to .066 seconds additional when a STEP is specified. If one of the variables A to V is used as the control variable, the time required is .017 seconds longer.

The NEXT statement, when the mating FOR statement ends a line and when the control variable is W , X , Y or Z , takes .178 seconds for each cycle of the loop. If the FOR statement does not end a line, .014 seconds is added. When A through V serve as the control variable, .050 seconds is added.

Functions

The times noted for functions do not include the time required to store or recall the variable. Thus, the time shown for $\text{ABS } A$ is the difference in time between executing $W=\text{ABS } A$ and $W=A$.

$\text{ABS } A$.011 to .013 seconds.

$-A$.012 to .014 seconds.

$\text{SGN } A$.014 to .016 seconds.

$\text{INT } A$.016 to .019 seconds for A greater than zero.

.020 for $A = 0$.

.017 to .020 for negative integers.

.023 to .036 for negative non-integers. The additional time is used for subtracting 1.

$\sqrt{A}$.056 to .159 seconds. The time depends mainly on the sum of the digits in the resulting value.

$\text{LOG } A$ The time varies and depends primarily on the number of factors of the form 2, 1.1, 1.01, 1.001, etc., whose product equals the mantissa of A .

For $A>1$, the time is from .108 to about .390 seconds.

For $A<1$, from .035 to .108 additional seconds is required for the division performed in calculating $\text{LOG } A$ in the form $-\text{LOG}(1/A)$.

$\text{LN } A$ The time required is from .035 to .108 seconds longer than for $\text{LOG } A$. This is because of the division needed to calculate $\text{LN } A$ as $(\text{LOG } A)/.434294481903$.

10^A Although not directly accessible to the user, a routine that performs this calculation is used by the computer in calculating $\text{EXP } A$ and A^B .

The time to perform the function varies depending on the number of terms of the form $\text{LOG } 2$, $\text{LOG } 1.1$, $\text{LOG } 1.01$, $\text{LOG } 1.001$, etc., whose sum is the fractional part of A .

For negative values of A , $1/10^{-A}$ is calculated. This division operation adds some .035 to .108 seconds.

$\text{EXP } A$ This is calculated as $10^{A*.434294481903}$. When A is greater than zero, the time ranges from .129 to approximately .360 seconds. For $A=0$ the time is .107 seconds. For A less than zero, an additional .035 to .108 seconds is used.

Summary

With the timing information provided here, the sophisticated programmer can do a great deal to increase the operating speed of many routines.

A MESSAGE FOR AUTHORS

PCN invites the submission of quality programs for pocket computers as well as practical application and tutorial articles of specific interest to users of pocket computers.

PCN purchases all rights to accepted materials at rates ranging from \$50 to \$150 per finished page as it appears in *PCN*. The payment rate is highly dependent upon the editorial and technical quality, timeliness and general suitability of the material for our readers.

Please double-space all manuscript material. Include a clean printer listing and a tape cassette copy of substantial listings. Be sure and include a self-addressed, stamped envelope if you desire the return of material that has been submitted for our consideration.

We keep all materials, including cassette tapes, and provide an Author's Agreement when a submission meets our requirements.

Since all submitted material must be critically evaluated and reviewed, it may take three to six weeks before any decision regarding a submission is made.

PCN also welcomes letters to the editor that express opinions, provide operating tips, programming tricks or news and general information of interest to fellow users of pocket computers. Selected material from such letters may be published, without monetary remuneration, as a service to subscribers.

Please direct all submissions for editorial review to:

The Editor
POCKET COMPUTER NEWSLETTER
35 Old State Road
Oxford, CT 06483

SUPERMARKET COMPANION

Here are two programs for the harried supermarket shopper. Both are for the Radio Shack TRS-80 or Sharp PC-1211 pocket computers. The first version is for people who do not own or do not want to carry the printer unit into the store. The second version is for those who would like printed output.

Operating Instructions

After loading the program, place the computer in the DEF mode. If you are using the printer version, be sure you have the printer initialized by pressing the ON button twice. Press the SHIFT key and SPC to begin.

Enter the sales tax for your state. For instance, if it is 6%, then just enter 6. If it is 3.75%, then enter 3.75.

Enter the budget allowed for the current shopping trip. Once this has been done, the printer version will output a copy of the data inputted as you shop.

The computer will now display: ENTER AMOUNT. There are several options that may be elected in answering this query:

1. *Non-taxable Item.* Enter the amount of the item you are going to purchase. Example: A loaf of bread is \$1.29. You would enter 1.29.

2. *Taxable Item.* For taxable items, enter the amount as 1.29T. The computer then automatically figures the tax. How does it do that? No real mystery. During the initialization process the variable T was given the value of your tax. If you inputted 6% for your tax, then the variable T now holds the value 1.06. Remember, the computer pre-solves arithmetic before acting upon it in the program. You could answer

Program Supermarket Companion

```

05: REM "<1981> K. SLAUGHTER"
10: " "

[CLEAR & INITIALIZE BLOCK]

20: CLEAR
22: USING"#####.##"

25: E=10^-20

30: PAUSE "SUPERMARKET: COMP 1.0"
40: PAUSE "INITIALIZE. . ."
50: INPUT "SALES TAX (AS 6.5%)? ";X
60: IF X<0 PAUSE "WHOLE NUMBERS!";GOTO 50
70: IF X=0 PAUSE "WARNING. . .":
    PAUSE "SALES TAX. . .":
    PAUSE "NOT INITIALIZED":
    GOTO 100
80: T=1+(.01X)

90: INPUT "BUDGET LIMIT? ";B
95: IF B<0 BEEP 1:GOTO 90

[INPUT BLOCK]

100: X=0:INPUT"ENTER AMOUNT: ";X
110: IF X=0 GOSUB 200
120: IF X=E GOTO 300
130: Z=Z+X
140: R=B-Z
150: IF R<0 BEEP 1:
    PAUSE"WARNING. . .":
    PAUSE"OVER BUDGET!":
    GOSUB 200
190: GOTO 100

[SUBROUTINE: DISPLAY ACCUMULATED RESULTS]

200: PRINT"T=";Z;" R=";R:
    RETURN

300: FOR I=1 TO 3
310: PAUSE "END. . ."
320: NEXT I

330: GOSUB 200
340: END
    
```

Program Supermarket Companion — Printer Version

```

05: REM "<1981> K. SLAUGHTER"
10: " "

[CLEAR & INITIALIZE BLOCK]

20: CLEAR
22: USING"#####.##"

25: E=10^-20

30: PAUSE "SUPERMARKET: COMP/PTR 1.1"
40: PAUSE "INITIALIZE. . ."
50: INPUT "SALES TAX (AS 6.5%)? ";X
60: IF X<0 PAUSE "WHOLE NUMBERS!";GOTO 50
70: IF X=0 PAUSE "WARNING. . .":
    PAUSE "SALES TAX. . .":
    PAUSE "NOT INITIALIZED":
    GOTO 100
80: T=1+(.01X)

90: INPUT "BUDGET LIMIT? ";B
95: IF B<0 BEEP 1:GOTO 90

99: GOSUB 500

[INPUT BLOCK]

100: X=0:INPUT"ENTER AMOUNT: ";X
110: IF X=0 GOSUB 200:GOTO 100
120: IF X=E GOTO 300
125: M$="":INPUT"ITEM: ";M$
130: Z=Z+X
140: R=B-Z
150: IF R<0 BEEP 1:
    PAUSE"WARNING. . .":
    PAUSE"OVER BUDGET!":
    GOSUB 200

180: PRINT M$
185: PRINT X
190: GOTO 100

[SUBROUTINE: DISPLAY ACCUMULATED RESULTS]

200: PAUSE"TOTAL=";Z
210: PAUSE"REMAINING=";R
220: RETURN

[EXIT]

300: FOR I=1 TO 3
310: PAUSE "END. . ."
320: NEXT I
330: GOSUB 200

340: PRINT"*****"
350: PRINT"TOTAL=";Z
360: IF R<0 PRINT "OVER BUDGET":
    R=-R:
    PRINT R:
    GOTO 380
370: PRINT "UNDER BUDGET":
    PRINT R
380: PRINT " "
390: PRINT"—END"

395: FOR I=1 TO 3
396: PRINT " "
397: NEXT I
400: END

[PRINTER INITIALIZATION]

500: PRINT"PRINTER READY. . ."
510: PRINT " "
520: PRINT " * SUPERMARKET * "
530: PRINT " "
540: PRINT"X=";X;"%"
550: PRINT " "
560: PRINT"BUDGET= $";B
570: PRINT " "
580: RETURN
    
```

any numeric input question with a formula such as: 69×1.29 , if you desired. Likewise, the computer recognizes $1.29T$ as $1.29 \times T$ or, in this example, 1.29×1.06 .

3. *Press ENTER to Display Totals.* If you do not enter a value, the computer will display the accumulated total and the amount of your budget that remains. You may press ENTER as many times as you like.

4. *Enter 'E' to End.* To exit the data input loop and display final totals, just enter the letter E (followed by ENTER, of course). This operation works the same way that using the letter T does when inputting a price. The variable E was initialized to a value that is not likely to be entered as an item price. When the program finds this unlikely value it transfers control to the termination sequence.

5. *Enter a Credit.* If you want to reverse a charge, simply enter the value again as a negative number. If the item was taxable, don't forget to add the T suffix. Example: $-1.25T$.

If you use the printer version of the program, a second question termed ITEM is asked. You can thus enter a brief description (maximum of seven characters) of each purchase or credit.

If you go over budget, the computer warns you, but it will continue program execution.

Imagine the surprise on your checker's face when you hand over your itemized purchase slip!

Thanks for this nicely documented package go to: *Ken Slaughter, 2916 Bangor Avenue, Highland, CA 92346.*

TRIANGLES

When three parts of a triangle are given, including at least one side, this program determines the missing sides and angles, as well as the area of the triangle.

Operating Instructions

Make sure the computer is set to the DEGREE mode. Note that side A denotes the side opposite angle A, etc., in a triangle ABC. Execute the RUN command to start the program. Enter inputs as prompted. If the side or angle requested is not known, key ENTER alone.

An Example

Given side A=5, side C=8 and angle A=35 degrees:

SIDE A? 5
SIDE B? —
SIDE C? 8
ANGLE A? 35
ANGLE B? —
ANGLE C? —

The calculated results:

FIRST OF TWO SOLUTIONS:

SIDE A = 5.
SIDE B = 8.539329288
SIDE C = 8.
ANGLE A = 35.
ANGLE B = 78.40466339
ANGLE C = 66.59533661
AREA = 19.59183225
SECOND SOLUTION:
SIDE A = 5.
SIDE B = 4.567103421
SIDE C = 8.
ANGLE A = 35.
ANGLE B = 31.59533661
ANGLE C = 113.4046634
AREA = 10.47833162

Program Notes

Variables I and J are used to keep track of which sides and/or angles were given. If one side and two angles are given, the Law of Sines is used in line 20 — 22. If two sides and the angle opposite one of them are given, the Law of Sines is used in lines 30 — 35. If two sides and the included angle are given, lines 40 — 43 are used. Although the Law

of Cosines could be used here, the method used by the program gives improved accuracy in certain cases. When all three sides are given, the angles are found using the Law of Cosines in lines 50 — 51.

This program submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

Program Triangles

```

10: CLEAR : INPUT      -M: A(I+3)=M-
   "SIDE A? "IA       L: A(I)=G*SIN
   : I=1              (M-L)/SIN L:
11: INPUT "SIDE        N=0: PRINT "S
   B? "IB: I=I+2      ECOND SOLUTI
12: INPUT "SIDE        ON: "GOTO 60
   C? "IC: I=I+3-1    40: L=A(I+3): FOR
13: IF I=0 GOTO 5      K=4 TO 6
   0                   41: IF K<>I+3 LET
14: INPUT "ANGLE       G=A(K-3): H=A
   A? "ID: J=1         (9-I-K): A(K)
15: INPUT "ANGLE       =90: IF H<>G:
   B? "IE: J=J+        COS LLET A(K
   2                   )=ATN (G*SIN
16: INPUT "ANGLE       L/(H-G*COS L
   C? "IF: J=J+3-      42: IF -A(K)LET
   J                   A(K)=180+A(K
17: IF A(I)=0          )
   GOTO 30             43: NEXT K: A(I)=
20: L=90-D-E-F+9      JCGG*SIN L*
   0: IF L<=0          SIN L+(G*COS
   GOTO 70             L-H)*(G*COS
21: A(J+3)=L: FOR      L-H: GOTO 60
   K=1 TO 3: IF K      50: IF (A+B-C)*(
   <>I LET A(K)=        A+C-B)*(B+C-
   SIN A(K+3)*A        A)<=0 GOTO 70
   (I)/SIN A(I+3)     51: D=ACS ((BB+C
22: NEXT K: GOTO      C-AA)/2BC: E=
   60                  ACS ((AA+CC-
30: IF J=I GOTO 4      BB)/2AC: F=
   0                   ACS ((AA+BB-
31: G=A(J): H=A(C     CC)/2AB
   -I-J): L=A(J+3): IF H*SIN
   L>G GOTO 70         60: K=.5AB*SIN F
32: M=ASN (H*SIN      : PRINT "SIDE
   L/G: A(9-I-J)       A= "IA:
   =M: IF L+M>=1       PRINT "SIDE
   80 GOTO 70          B= "IB: PRINT
33: A(I+3)=180-L      "SIDE C= "IC
   -M: A(I)=G*         61: PRINT "ANGLE
   SIN (L+M)/          A= "ID:
   SIN L: IF G>H      PRINT "ANGLE
   *SIN L IF H>G      B= "IE:
   PRINT "FIRST        PRINT "ANGLE
   OF TWO SOLU        C= "IF:
   TIONS: "N=1         PRINT "AREA=
34: GOTO 60           "IK: IF N
35: A(9-I-J)=180      GOTO 35
                     62: END
                     70: PRINT "NO SO
                     LUTION"

```

CORRECTION TO TIME ADDITION PROGRAM

Alan B. Owens was the first alert reader to point out that line 150 in the Time Addition program printed on page 6 of PCN Issue 10 should read:

150 B=10000*DMS B

Apparently a zero was dropped in the program listing originally sent to PCN.

A PROGRAM FOR AVIATORS

The accompanying program is designed to take care of the tedious calculations that aviators perform when plotting a flight plan. The program is specifically designed for a Radio Shack TRS-80 or Sharp PC-1211 and should be operated in the DEF mode. Follow these steps:

1. Press SHIFT/A. The computer will ask for the variance from magnetic north. Input the variance in degrees with the appropriate sign. Use a minus sign for east variations. West variations should be entered as a positive value. Press ENTER after keying the data.
2. Input the wind direction as requested.
3. Input the wind velocity in statute miles-per-hour. Since most flight services report wind velocity as knots, a separate conversion routine is provided. Use SHIFT/B to convert knots-per-hour to statute miles-per-hour. (This should be done prior to starting this program.)
4. Enter the true air speed at which you plan to fly your craft.
5. Enter the fuel burn rate for your ship in gallons-per-hour.
6. Enter the true course plotted.
7. Enter the distance of the flight (or flight leg) in statute miles.
8. The computer will output: the wind correction angle (WCA), the true heading at which to fly your ship to account for the wind, the magnetic heading, the projected ground speed in miles-per-hour, the estimated enroute time and the predicted fuel usage.
9. Respond appropriately to the query from the computer concerning calculations for a new flight leg.
10. If you use a printer, all the output will be printed at one time. If you use the PC alone, you must press ENTER after each computed value has been displayed.

This program was submitted by: *Gene Robertson, 5757 Southwestern Boulevard, Hamburg, NY 14075.*

Program Aviation

```

10:"A"
20:PAUSE "FLIGHT DATA"
30:INPUT "VARIANCE(E-W+)",A
40:INPUT "WIND DIRECTION",B
   "WIND VELOCITY",C
50:INPUT "T.A.S.",D,"BURN",E
60:INPUT "TRUE COURSE",F
70:INPUT "DISTANCE",G
80:H=ASN((C/D)*SIN(B-F))
90:PRINT "W.C.A.",H
100:I=H+F
110:PRINT "TRUE HEADING",I
120:J=I+A
130:PRINT "MAG. HEADING",J
140:K=D*COS(I-F)-C*COS(B-F)
150:PRINT "GROUND SPEED",K
160:L=G/K
170:M=L*60
180:PRINT "TIME ENROUTE",M
190:N=E*L
200:PRINT "FUEL USED",N
210:INPUT "NEW LEG?(YES-NO)",A$
220:IF A$="YES" GOTO 30
230:PRINT "HAVE A GOOD FLIGHT"
240:END
250:"B"
260:PAUSE "KNOTS TO MPH"
270:INPUT "ENTER KNOTS",O
280:P=O*.67
290:PRINT P," M.P.H."
300:END

```

LINEAR REGRESSION ANALYSIS

When N pairs (X,Y) of numbers are entered, the computer displays the slope and Y-intercept of the *regression line* (also known as the *least-squares* or *trend line*). Other relevant statistical information is also provided.

An Example of Program Operation

Given the pairs (15,37), (18,40), (17,40), (22,46) and (20,48). Start the program by giving the RUN directive. Respond to program prompts as follows:

```

NO OF PAIRS? 5
X? 15
Y? 37
X? 18
Y? 40
X? 17
Y? 40
X? 22
Y? 46
X? 20
Y? 48

```

Information is then provided by the program as follows:

```

REGRESSION LINE:
SLOPE= 1.52739726
Y-INT = 14.09589042

```

Therefore, the equation of the regression line is approximately:

$$Y = 1.527X + 14.096$$

The program continues to yield:

```

VAR.= 5.559360731
STD DEV= 2.357829665

```

These are the variance and standard deviation of the errors, using the sample values. The standard deviation is also known as the *standard error of estimate*. Note: the program calculates these as the *unbiased* variance and standard deviation.

The program yields still more:
CORR.= 8.962837176 E-01

Thus, the coefficient of correlation in the example is .896. The program will now show:

```

SIG LEV FOR LINEARITY,
1.97333026 E-02

```

This may be interpreted as meaning that if there is no linear relationship between X and Y, the probability of a sample such as the one used here for illustration, would be only about .0197. Customarily, a probability less than .05 is considered as sufficient evidence that a linear relationship exists. At this point, if the variable T were displayed, it would be 3.500507443. This is the *Student's T*. With N-2 degrees of freedom, which is 3 in this case, the probability of a value of T this large is .097.

The program offers still more:

```

95% C.I. FOR SLOPE:
MIN, 0.13878151
MAX, 2.91601301

```

The preceding shows a 95 percent confidence interval for the slope of the *population* regression line.

There is still more:

```

95% CI FOR CORR COEFF:
MIN, 6.69917843 E-02
MAX, 9.931811736 E-01

```

The *population* coefficient of correlation is thus most likely between .067 and .993 for the example data.

Now, predictions of Y for a given X may be calculated. Press the ENTER key one more time:

```

GIVEN X? 16 (This number chosen merely as an example.)
Y= 38.53424658

```

Thus, with X=16, the regression-line value of Y is about 38.534. You can get more information about this prediction. Press ENTER again:

```

95% CI FOR MEAN Y:
MIN, 33.80479176
MAX, 43.2637014
95% CI FOR PREDICTED Y:
MIN, 29.66448046
MAX, 47.4040127

```

```

10: INPUT "NO OF
    PAIRS?" N: A
    =0: B=0: C=0: D
    =0: E=0: FOR Z
    =1 TO N
11: INPUT "X? " X
    "Y? " Y: A=
    A+X: B=B+Y: C=
    C+X: D=D+Y:
    E=E+Y: NEXT
    Z
12: F=CN-AA: G=DN
    -AB: H=EN-BB:
    K=N-2: R=G/JF
    H: V=(FH-GG)/
    FKH: S=IV: M=G
    /K: L=(B-AM)/
    H
13: PRINT "REGRE
    SSION LINE:"
    : PRINT "SLOP
    E=" S: PRINT
    "Y-INT =" L
    : PRINT "VAR.
    =" V: PRINT
    "STD DEV=" S
14: PRINT "CORR.
    =" R
15: T=ABS M/(C/
    NV(X)=K/(K+T
    *P=K+1: Y=P
16: IF K>2 FOR Y=
    K-3 TO 2 STEP
    -2: P=1+PXY/(
    Y+1: NEXT Y
17: IF Y=1 LET P=
    .5PT/J(K+TT:
    GOTO 24
18: P=PTX/TK/TT+
    ATN (T/TK)/1
    80
19: P=.5-P: BEEP
    1: PRINT "SIG
    LEV FOR LIN
    EARITY:"
    PRINT P
20: IF K<8 GOTO 3
    2+K
21: T=((K+.17/K+.
    578)/K+.7359
    )/K+1.53953)
    /K+2.53585
22: T=((T/K+2.82
    24966)/K+2.3
    7227123)/K+1
    .959963985:
    GOTO 40
23: T=TAN 85.5:

```

```

GOTO 40
24: T=38/J78:
    GOTO 40
25: T=3.18244630
    5: GOTO 40
26: T=2.77644510
    5: GOTO 40
27: T=2.57058183
    6: GOTO 40
28: T=2.44691185
    1: GOTO 40
29: T=2.36462425
    2
40: Y=T*(CNV/F: X
    =M-Y: Y=M+Y:
    PRINT "95% C
    I. FOR SLOP
    E:" PRINT "M
    IN:" X:
    PRINT "MAX:
    "Y
41: IF N=3 GOTO 5
    0
42: Y=LN ((JFH+G
    )/(JFH-G: Z=3
    .919927969/J
    (N-3): X=(EXP
    (Y-Z)-1)/(1+
    EXP (Y-Z)
43: Y=(EXP (Y+Z)
    -1)/(1+EXP (
    Y+Z: PRINT "9
    5% CI FOR CO
    RR COEFF:"
    PRINT "MIN:
    "X: PRINT "M
    AX:" Y
50: INPUT "GIVEN
    X?" X: Y=MX+
    L: PRINT "Y="
    "Y: U=X-A/N:
    U=NUU/F+1/N
51: Z=T+JUV: I=Y-
    Z: J=Y+Z:
    PRINT "95% C
    I FOR MEAN Y
    ":" PRINT "MI
    N:" J: PRINT
    "MAX:" J
52: Z=T*(J(U+1): V:
    I=Y-Z: J=Y+Z:
    PRINT "95% C
    I FOR FREDIC
    TED Y:"
    PRINT "MIN:
    "J: PRINT "M
    AX:" J: GOTO
    50

```

This information may be interpreted as follows:

When X=16, the "best guess" for Y is 38.534. When X=16 the mean value of Y should be between 33.80 and 43.26. In an individual case, Y may be expected to fall between 29.66 and 47.40.

The value of Student's T being used in determining these confidence

intervals is 3.182446305, which is $t_{.025}$ for 3 degrees of freedom, correct to about 10 digits. This value may be observed by displaying the variable T. (It is calculated in line 30 through 39 of the program.)

This program submitted by: Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.

ANY BASE CONVERSION PROGRAM

Emerich Auersbacher, 41 King Street - Apt. 2, Belleville, NJ 07109 provided this program that converts any number (fractional parts, too) in base A to its equivalent in base B. The bases A and B can range from 2 to 100. A special feature of this program is that the number to be converted is inputted in its entirety instead of digit by digit. This makes the program easy to use. (The secret to this capability is a "parser" in the program that separates the digits and examines them one at a time!)

Note: When the base is in the range 11 to 100, two positions are allocated to each "digit" (symbol). Make sure you remember this when interpreting results.

[Thanks for this version of a program that a number of readers had requested. Your parsing procedure provides a special touch of elegance to its practical application. — N.W.]

Program Any Base Conversion

```

2: PAUSE " BASE
    CONVERSION
    PROG." USING
4: CLEAR: INPUT
    "KNOWN NUMBE
    R:" X, "BASE
    OF NUMBER:"
    B, "BASE DESI
    RED:" J
6: IF P=10 LET W
    =X: GOSUB 20:
    B=P: GOTO 12
8: IF B=10 LET S
    =X: V=S: B=P:
    GOSUB 180: Y=
    N: GOTO 12
10: W=X: GOSUB 20
    : S=Y: V=S: B=P
    : GOSUB 180: Y
    =N
12: BEEP 3: PAUSE
    "ANSWER >>—
    >"
14: PRINT "A=" IY
    : " BASE " JB
16: GOTO 4
20: N=0: F=10: IF
    B>10 LET F=B:
25: IF X<1 GOTO 9
    0
30: T=X/(10^N):
    IF T<.1 GOTO
    50
40: N=N+1: GOTO 3
    0
50: IF N=0 GOTO 9
    0
55: S=INT (X/(F^
    (N-2)))
58: IF N<=2 GOTO
    90
60: FOR Z=1 TO N-

```

```

2
65: D=INT (X/(F^
    (N-2-Z))) -F*
    INT (X/(F^(N
    -1-Z)))
70: S=(S*B)+D
80: NEXT Z
90: Y=0: IF X=INT
    X=0 GOTO 150
95: X=X-INT X
100: N=1
105: V=X/F^N: IF V
    -INT V=0 GOTO
    125
110: N=N+1: GOTO 1
    05
125: FOR Z=1 TO N
126: IF Z=1 LET D=
    INT (V/(F^(N
    -1))) : GOTO 1
    35
130: D=INT (V/(F^
    (N-Z))) -F*
    INT (V/(F^(N
    -Z+1)))
135: Y=Y+D*B^(N-Z)
140: NEXT Z
150: Y=Y+S: RETURN
160: IF 10=B LET
    R=10
185: IF B>10 LET R
    =100
190: N=0: T=0
195: IF S<=0 GOTO
    220
200: W=LN S/LN B
205: W=INT (W)
210: N=N+R^W: IF T
    =0 GOTO 220
215: S=S-B^W: T=N:
    GOTO 195
220: RETURN

```

AN ELECTRONIC CARD DECK

Emerich Auersbacher, 41 King Street - Apt. 2, Belleville, NJ 07109, sent in this combination card shuffling and dealing program. To "draw" a card at random from the "electronic deck," just press the ENTER button. (After, of course, you have started the program with the RUN command.) Continue pressing the ENTER key to get as many cards dealt as needed. Once a card has been displayed, it will not show up again unless the deck is reshuffled. The program may be used as is to play a number of card games. Alternately, you might want to use the program as a subroutine in a game of your own design. Note that the program displays a card by both rank and suit such as:

CARD: J OF CLUBS

Program Cards

```

5: USING :BEEP
1: PAUSE " E
  ELECTRONIC CA
  RD DECK":S=5
3
10: A$="DIAMOND"
  B$="HEART",
  C$="CLUB", D$
  ="SPADE"
20: INPUT "PICK
  ANY #":G=G+
  ABS (439147+
  EG*G+G):H=E8
  +1
30: IF S>=42:BEEP
  1: PAUSE "SHU
  FFLE DECK":S
  =0:GOSUB 100
40: I=23G:G=I-
  INT (I/H)*H,
  R=INT (52*G/
  H)+1
45: IF A$(20+R)=
  " ":GOTO 40
50: S=S+1,T=R/4,
  T=INT ((T-
  INT T)*4)+1
55: A$(5)=A$(20+
  R):A$(20+R)=
  " "
60: BEEP 1:PRINT
  "CARD: "A$
  (5)" OF "A$
  (T)"S"
70: GOTO 30
100: FOR J=21 TO 2
  4:A$(J)="A":
  NEXT J:FOR J
  =25 TO 28:A$(
  J)="2":NEXT
  J:FOR J=29 TO
  32:A$(J)="3"
  :NEXT J
  101:FOR J=33 TO 3
  6:A$(J)="4":
  NEXT J:PAUSE
  "SHUFFLING":
  FOR J=37 TO 4
  0:A$(J)="5":
  NEXT J
  102:FOR J=41 TO 4
  4:A$(J)="6":
  NEXT J:FOR J
  =45 TO 48:A$(
  J)="7":NEXT
  J:FOR J=49 TO
  52:A$(J)="8"
  :NEXT J
  103:FOR J=53 TO 5
  6:A$(J)="9":
  NEXT J:PAUSE
  "SHUFFLING":
  FOR J=57 TO 6
  0:A$(J)="10"
  :NEXT J
  104:FOR J=61 TO 6
  4:A$(J)="J":
  NEXT J:FOR J
  =65 TO 68:A$(
  J)="Q":NEXT
  J:FOR J=69 TO
  72:A$(J)="K"
  :NEXT J
  105:RETURN

```

OPERATING TIP SPEEDS INSERTION OF NEW STATEMENTS

You can store a string of dashes using a reservable key, for instance:
SHIFT SPC: -----

When you want to insert something new in an existing line, key in a SHIFT INSert at the appropriate point, then hit SHIFT SPC to drop in the string of dashes. Key in the new characters right over the dashes and eliminate unused dashes with spaces. Submitted by *George Fergus*.



35 Old State Rd, Oxford, CT 06483

FROM THE WATCH POCKET

Sinclair Research Limited reports selling over a quarter of a million of their ZX81 personal computers in less than a year. Sales of it and its predecessor, the ZX80, total more than 400,000 units. While the unit doesn't quite fit in a pocket, it is pretty small and weighs in at just 12 ounces. Apparently a lot of PCN readers are interested in the unit. Of course, a lot of the routines presented herein are readily adaptable to the ZX80 or ZX81. For more information on the ZX81 write directly to Sinclair at 50 Staniford Street, Boston, MA 02114. The phone number is (617) 742-4826. The unit is only sold by mail order at the present time in the United States. They are doing just fine, though, moving some 4,000 units per week!

It didn't take long for my prediction of last month to come true. The Sharp PC-1500 represents a new generation of PC. The price range of \$300.00 is sure to please many. Yes indeed, 1982 is going to be an exciting year for pocket computer enthusiasts!

By the way, if you want to be among the first to get your hands on a PC-1500 in the U.S., better get a call in to *Mort Rosenstein* at Atlantic N.E. Marketing. The phone number is (617) 639-0285. He indicates they are scheduled to get some of the first units released. However, all he can do right now is reserve a unit in your name.

I have received several firm reports that Hewlett-Packard is circulating some pre-production versions of a HHC. This unit sounds intriguing with reports that it will be able to handle 32K of memory. HP has also developed a system that will enable its HHCs and even some of its calculators to interface directly with its larger computers. The portable units can thus acquire data in the field, dump it into a larger machine for processing, and retrieve that information to maintain a current, condensed data base.

Programs for pocket computers fared much better in the popular trade press the past month. 80 Microcomputing has a whole slew of PC programs in the December, 1981 issue. They include a Metric/English Conversion program, a Depreciation Analysis routine, a program that calculates Loan Payment/Interest, another that performs Investment Calculations and a program you can use on visits to the supermarket called Sharp Marketing. While most of these types of applications have been covered in previous issues of PCN, you might want to take a look at another programmer's approach.

Interface Age has several PC articles in its January, 1982 issue. *Bob McElwain* has a well documented version of a game called Bagles Plus One. *David D. Busch* presents a program that keeps track of gasoline usage, miles-per-gallon performance and other statistics when on an automobile trip.

Cass R. Lewart has a PC program in the December 15, 1981 issue of *Electronics*, page 159, that determines suitable scales for computer-generated plots.

Missouri Indexing, Inc., 7 Watch Hill Road, St. Louis, MO 63124, has published a reference called *The Index*. There are some 30,000 entries in it covering some 6 years of articles, editorials and columns from over 800 issues of 45 computer-related publications. The reference was compiled by *William H. Wallace*, an attorney. It utilizes a method of indexing called *Key Word in Context*. Using it you can rapidly locate information for particular types of computers. For pricing and additional information write to the publisher at P.O. Box 301, St. Ann, MO 63074. The phone number is (314) 997-6470.

It is interesting to note that Panasonic is advertising their HHC as *The Link* in magazines such as *Fortune*. As yet, I have not seen them advertising it in the personal computer journals.

In response to requests from readers, we are planning on having some hardware-related articles in the near future. How would you like to use your PC as a simple controller? We have located a clever electronic designer who has done amazing things with a PC in this regards. I hope to be able to provide you with the details of his exciting work in the near future.

Since this is the start of the second year of publishing PCN, I would like to welcome all the new subscribers who are coming aboard with this issue as well as thank the high percentage of Charter Subscribers who are staying with us.

Best wishes for a successful 1982!

— *Nat Wadsworth, Editor*